



# **Utveckling av en responsiv nätsida för HPTY Recodings**

Totte Blomqvist

Förnamn Efternamn

Examensarbete

Medieteknik

2016

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Medieteknik
Identifikationsnummer:	
Författare:	Totte Blomqvist
Arbetets namn:	
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	Jaako Hurme
<p>Sammandrag:</p> <p>I dag är det ganska populärt att grunda ett eget skivbolag för att ge ut sin egen musik. Det finns otaliga skivbolag som man inte vet att existerar överhuvudtaget. För att bättre kunna nå de människor som vet om skivbolaget och för att få mera synlighet är det bra att ha en webbplats. Det var tanken när jag fick uppdraget att utveckla en webbplats för HPTY Recordings.</p> <p>I dag surfar man på webben ofta med en pekplatta och smarttelefon. För att webbplatsen också skall fungera på surfplattor och smarttelefoner måste den vara responsiv. D.v.s. webbplatsen skall ha samma funktionalitet och se bra ut oberoende med vilken apparat man besöker webbsidan. I detta examensarbete kommer jag att berätta om webbsidans utvecklings- och designprocessen, samt redogöra vad responsivitet innebär och hur det i praktiken går till att göra en webbsida responsiv. På grund av att det finns så många olika sätt att göra en webbsida responsiv har jag valt några allmänna metoder för noggrannare granskning och skall försöka lista ut vilken metod passar bäst just för denna webbsida.</p>	
Nyckelord:	Responsiv webbdesign, RWD, Adaptiv webb design, AWD, HPTY Recordings, Design, Programmering
Sidantal:	
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Mediatechnology
Identification number:	
Author:	Totte Blomqvist
Title:	Development of a responsive website for hpty recordings
Supervisor (Arcada):	Johnny Biström
Commissioned by:	HPTY Recordings
<p>Abstract:</p> <p>Today it is quite popular to start a record label to release your own music on. There are countless small record labels that only a handful of people know that they even exist. To setup a website is a great way to reach your followers and gain exposure. That was the main idea when I got the assignment to develop a website for HPTY Recordings.</p> <p>Now days it is very common to browse the web with a smartphone or a tablet. In order to work properly on smaller displays the website needs to be responsive. In other words, the website should have the same functionality and look good regardless of what kind of a device you are using. In this thesis, I will tell you about the web page development and design process, and describe what responsivity is and how a website is made responsive. Because there are so many different ways to make a website responsive, I have chosen a couple methods for closer examination and will try to figure out which method is most suitable for this website.</p>	
Keywords:	Responsive web design, RWD, Adaptive web design, AWD, HPTY Recordings, Design, Development
Number of pages:	
Language:	Swedish
Date of acceptance:	

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Mediatekniikka
Tunnistenumero:	
Tekijä:	Totte Blomqvist
Työn nimi:	Responsiivisten nettisivujen kehittäminen HPTY Recordingsille
Työn ohjaaja (Arcada):	Johnny Biström
Toimeksiantaja:	HPTY Recordings
<p>Tiivistelmä:</p> <p>Nykyisin on aika yleistä perustaa oma levy-yhtiö julkaistakseen itse tuottamaa musiikkia. Maailmassa on lukuisia levy-yhtiöitä joista vain pieni joukko ihmisiä ovat tietoisia. Nettisivut on hyvä tapa lisätäkseen näkyvyyttä ja mahdollistaakseen suuremman yleisön tavoittamisen. Saadessani tehtäväksi kehittää nettisivut HPTY Recordingsille, tämä oli pääajatuksena.</p> <p>Nykyään käytetään yhä useammin netin selaamiseen joko taulutietokonetta tai älypuhelinia. Jotta nettisivut toimivat ja näyttävät hyvältä myös älypuhelimissa ja taulutietokoneissa on sivujen oltava responsiiviset. Toisin sanoen nettisivujen on toimittava kaikilla laitteilla. Tässä opinnäytetyössä käsittelen nettisivujen ulkoasun suunnittelu- ja ohjelmointi prosessia, sekä selvennän mitä responsiivisuus tarkoittaa ja miten responsiivisuus käytännössä saavutetaan. Koska menetelmiä on niin monia valitsin muutaman menetelmän lähempää tarkastelua varten ja yritän selvittää mikä menetelmä on paras juuri tälle nettisivuille.</p>	
Avainsanat:	Responsiivinen suunnittelu, RWD, Adaptiivinen suunnittelu, AWD, HPTY Recordings, Suunnittelu, Ohjelmointi
Sivumäärä:	
Kieli:	Ruotsi
Hyväksymispäivämäärä:	

# INNEHÅLL / CONTENTS

<b>1</b>	<b>Inledning.....</b>	<b>8</b>
1.1	Syfte .....	9
1.2	Avgränsning.....	9
1.3	Definitioner och förkortningar .....	10
<b>2</b>	<b>Planering.....</b>	<b>10</b>
2.1	Designen .....	11
2.1.1	<i>Design förslag.....</i>	<i>12</i>
<b>3</b>	<b>Responsive och adaptive web design.....</b>	<b>16</b>
3.1	RWD .....	17
3.1.1	<i>Flexbox layout .....</i>	<i>19</i>
3.1.2	<i>Bootstrap 3 .....</i>	<i>21</i>
3.1.3	<i>Restive.js .....</i>	<i>22</i>
<b>4</b>	<b>Genomförandet .....</b>	<b>25</b>
4.1	Verktyg .....	25
4.1.1	<i>Hastighets test.....</i>	<i>25</i>
4.2	Diskussion och motiveringar .....	26
	<b>Källor .....</b>	<b>30</b>

## Figurer

Figur 1. HPTY logo med svart text, vit bakgrund .....	13
Figur 2. HPTY logo med vit text, svart bakgrund .....	13
Figur 3. HPTY logo med ljusblå text, svart bakgrund. Logo alternativ 3. Texten "DO RIGHT SINCE 2007" härstammar från tiden före skivbolaget existerade och HPTY var ett radioprogram och klubb som ordnades på olika klubbar i Helsingfors.....	14
Figur 4. Kod för "Hem" knapp .....	14
Figur 5. Slutliga designen på startsidan. Footern rymdes inte med på bilden.....	15
Figur 6. Designen av subsidorna, endast iframens innehåll ändras .....	16
Figur 7. Ett modal fönster som skulle innehålla information av skivan och en musikspe- lare. Ett förslag som jag hade .....	16
Figur 8. Kod för modalen .....	17
Figur 9. En media query från HPTYs nätsida .....	19
Figur 10. CSS för flexbox exempel .....	21
Figur 11. Parent container och sex child items.....	21
Figur 12. När displayn blir mindre flyttar parent container på child items .....	22
Figur 13. Ett exempel hur en CSS3 fil kan se ut när man använder Restive.js. (Hämtad 20.4.2016. <a href="https://speckyboy.com/2014/04/24/make-any-website-responsive/">https://speckyboy.com/2014/04/24/make-any-website-responsive/</a> ) .....	23
Figur 14. Hur ett exempel skulle se ut med media queries. (Hämtad 20.4.2016. <a href="https://speckyboy.com/2014/04/24/make-any-website-responsive/">https://speckyboy.com/2014/04/24/make-any-website-responsive/</a> ) .....	24
Figur 15. Hälften av CSS:n som behövs för skriva media queries för alla storleks display. (Hämtad 20.4.2016. <a href="https://speckyboy.com/2014/04/24/make-any-website-responsive/">https://speckyboy.com/2014/04/24/make-any-website-responsive/</a> ) .....	24
Figur 16. CSS som krävs för att webbsidan skall vara responsive på alla olika storleks display, t.o.m. både i portrait och landscape. ((Hämtad 20.4.2016. <a href="https://speckyboy.com/2014/04/24/make-any-website-responsive/">https://speckyboy.com/2014/04/24/make-any-website-responsive/</a> ) .....	25
Figur 17. En funktion där det finns brytningspunkterna och namne av klasserna som man vill att skall användas vid dessa brytningspunkter.(Hämtad 20.4.2016 <a href="http://docs.restive.js.com">http://docs.restive.js.com</a> .....	25
Figur 18. Resultat av hastighetstest för startsidan .....	27
Figur 19. Hastighetstest för en subsida.....	27

Figur 20. Webbssidan på laptop.....	30
Figur 21. Webbssidan på en iphone5 och ipad .....	30

# 1 INLEDNING

I dagens läge är det nästan så att om ett bolag inte har en hemsida, existerar inte själva bolaget. Det finns otaliga små skivbolag som bara en handfull människor vet att existerar. Genom en webbplats kan man nå människor runt hela världen och få mera synlighet. Som examensarbete fick jag möjligheten att utveckla en hemsida för ett litet skivbolag, HPTY Recordings, som är baserat i Helsingfors. Min uppgift var att designa och skriva koden för hemsidan. HPTY Recordings har hittills klarat sig utan en egen hemsida för att verksamheten har varit så liten och mest hobby-baserat. Jaako Hurme och Henri Ekholm grundade HPTY Recordings år 2011 för att publicera musik som de själv producerat. De har publicerat i genomsnitt en skiva per år. Nu har verksamheten vuxit och HPTY har börjat publicera andra artisters musik också. Jag har bekantat mig med deras musik redan 2011 då de gav ut sin första skiva och jag har följt med deras publikationer sen dess. På hösten 2015 försökte jag hitta HPTY Recordings hemsida via google.com men hittade inte. Då kontaktade jag Jaako Hurme via Facebook.com och frågade om de skulle vara intresserade av att skaffa en hemsida för HPTY. Svaret var ja och vid årsskiftet 2016 satte vi igång.

Kundens krav var att hemsidans design skulle vara simpel och att det skulle vara enkelt att navigera på sidan. Grund tanken var att deras musik skulle spela huvudrollen på sidan. Från början var det meningen att sidan också skulle innehålla lite information om skivbolaget och dess artister m.m. men kunden ändrade sig och slopade iden.

Idag kopplar människorna sig till nätet allt oftare med en surfplatta eller en smarttelefon. Därför måste jag också anpassa siduppbyggnaden för dessa apparater, dvs. hemsidan skall vara responsiv. En nätsida som inte anpassar sig för en smarttelefon eller en surfplatta är jobbig att läsa och ofta nästan oanvändbar med andra apparater än en bords- eller bärbar dator. Det finns olika metoder att lösa problemet och i arbetet behandlar jag några olika metoder som jag funderade över att använda för att tackla problemet och vilken metod skulle vara bäst för just denna hemsida. Jag kommer också att berätta om hela processen att utveckla och designa en hemsida.



Uppgiften för mig var intressant och utmanande, precis vad jag ville att mitt examensarbete skulle vara. Jag har aldrig tidigare designat och utvecklat en nätsida helt från början till slut. Att göra nätsidan responsiv gjorde uppgiften ännu mer intressant för att det var ett okänt område för mig. Genom att anta utmaningen kunde jag också mäta mina egna färdigheter gällande kodandet och design.

## 1.1 Syfte

Mitt syfte på arbetet var att utveckla en responsiv hemsida med snygg design för kunden och att försöka hitta en bra lösning till detta, genom att jämföra några olika metoder att göra en nätsida responsiv. Kundens tillfredsställelse var mycket viktigt för mig.

Detta var första gången jag fick använda mina kunskaper inom programmering för ett professionellt ändamål. Samtidigt hade jag möjlighet att utveckla mina egna kunskaper under arbetsgången. På grund av att jag inte har tidigare erfarenhet av att utveckla en responsiv webbsida måste jag studera i de olika metoderna, hur man gör en websida responsiv. Största delen av informationen fick jag genom internet sökning och boken *Responsive web design* av Ethan Marcotte.

## 1.2 Avgränsning

Jag skall berätta om responsiv web design och vad det går ut på. Jag kommer att ta upp några olika metoder hur man gör en nätsida responsiv. Jag kommer också att mycket ytligt ta upp adaptive web design och berätta hur det skiljer sig från RWD.

## 1.3 Definitioner och förkortningar

**RWD Responsive web design** är en webbutvecklings strategi som gör att webbplatsen kan anpassa sig enligt användarens skärmstorlek.

**AWD Adaptive web design** är samma sak som responsive web design men använder andra metoder för att anpassa webbsidan för användarens skärm.

**CSS (Cascading Style Sheet)** är ett språk som formaterar layouten på webbsidor

**HTML (Hypertext Markup Language)** är ett märkspråk som används för att skapa dokument på World Wide Web. Används för att skapa webbsidor.

**PLUGIN/ RAMVERK** är som ett bibliotek med färdigt utvecklade funktioner. Kan också innehålla färdigt skriven CSS kod.

**MEDIA QUERY** är en kod snutt som bestämmer vilka CSS regler webbläsaren skall följa i vilken situation.

**JAVASCRIPT** är ett dynamiskt skriptspråk som används för att skapa funktionalitet på webbsidor.

**PARENT CONTAINER** är en yta på webbsidan där man placerar innehåll i.

**CHILD ITEMS** är innehåll som finns inne i parent containern.

## 2 PLANERING

För att arbetet skall framskrida smidigt måste det planeras bra i förväg. När det gäller ett beställningsarbete är kommunikationen mellan kunden och arbetstagaren mycket viktig.

På grund av att detta var mitt första arbete för någon annan än mig själv var det ett bra tillfälle att lära sig om kommunikationen mellan arbetstagare och kund. Första steget var att ordna ett möte där jag fick instruktioner om vad hemsidan skulle innehålla, hurdan färgtema kunden ville ha och hurdana funktioner sidan skall ha.

Vi kom överens om att jag gör några förslag av sidans struktur och layout enligt kundens guidelinjer och sedan byggde vi framåt och gjorde ändringar från det. Sidans innehåll har krympt mycket från vårt första möte. Ursprungligen skulle nätsidan innehålla bifogade bilder från instagram.com, bilder på skivbolagets skjortor som är till salu, och lite bakgrunds fakta om bolaget och dess artister. Kunden tyckte dock att det är musiken och publikationerna som skall spela huvudrollen och ville ha nätsidan så simpel som möjligt. För att undvika onödigt arbete hade vi möten med jämna mellanrum, men största delen av kommunikationen skedde via Facebook.com och per telefon.

## **2.1 Designen**

Att designa webbsidor var ganska nytt för mig. Jag har gjort två grafisk design kurser i skolan där jag lärde mig grunder och att använda olika mjukvara, som Adobes Photoshop CS5 och principer gällande design. Ofta görs designen först med ett bildredigeringsprogram. Det här var inte ett sådant arbete. Musiken som HPTY Recordings publicerar spelas oftast på mörka klubbar och i gamla övergivna lager. Det skulle på något vis komma fram i designen. Nätsidan skall alltså inte se ut som de flesta nätsidorna på nätet.

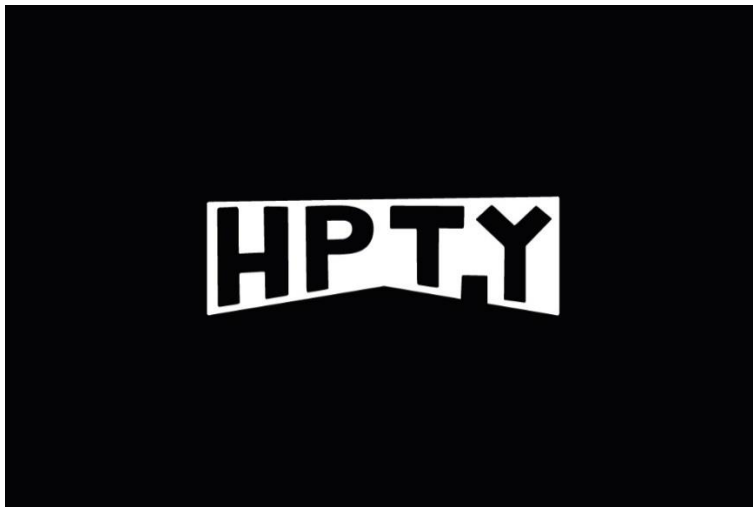
Jag fick några webbadresser av kontaktpersonen som han gillade och skulle fungera som referens. Sushitech.com var den som gav mig mest inspiration gällande designen. Jag hade också lite bildmaterial till mitt förfogande. Bland dem hittade jag bilden som fungerar som bakgrundsbild för webbsidan.

### 2.1.1 Design förslag

Bakgrundsbilden blev fastslagen med detsamma. Den behövde ingen redigering, bara klippas lite mindre. Det var klart att HPTY-logon ska finnas på webbsidan. Jag hade tre olika alternativ att välja mellan.



*Figur 1 HPTY logo med vit text, svart bakgrund. Logo alternativ 1.*



*Figur 2 HPTY logo med svart text, vit bakgrund. Logo alternativ 2.*



Figur 3 HPTY logo ljusblå text med svart bakgrund. Logo alternativ 3. Texten "DO RIGHT SINCE 2007" härstammar från tiden före skivbolaget existerade och HPTY var ett radioprogram och klubb som ordnades på olika klubbar i Helsingfors.

Jag tyckte mest om alternativ nr 3, men kontakt personen var inte av samma åsikt. Eftersom webbsidans bakgrund var så mörk var alternativ nr 2 den som passade bäst in. För att få logon att smälta in i bakgrunden bättre klippte jag bort överlopps yta från alla sidor. Resultatet var bra.

Logon skulle också fungera som en "Hem"-knapp.

```

15 <div class="logo">
16   <a href="www.index.html">
17     </a>
18 </div>

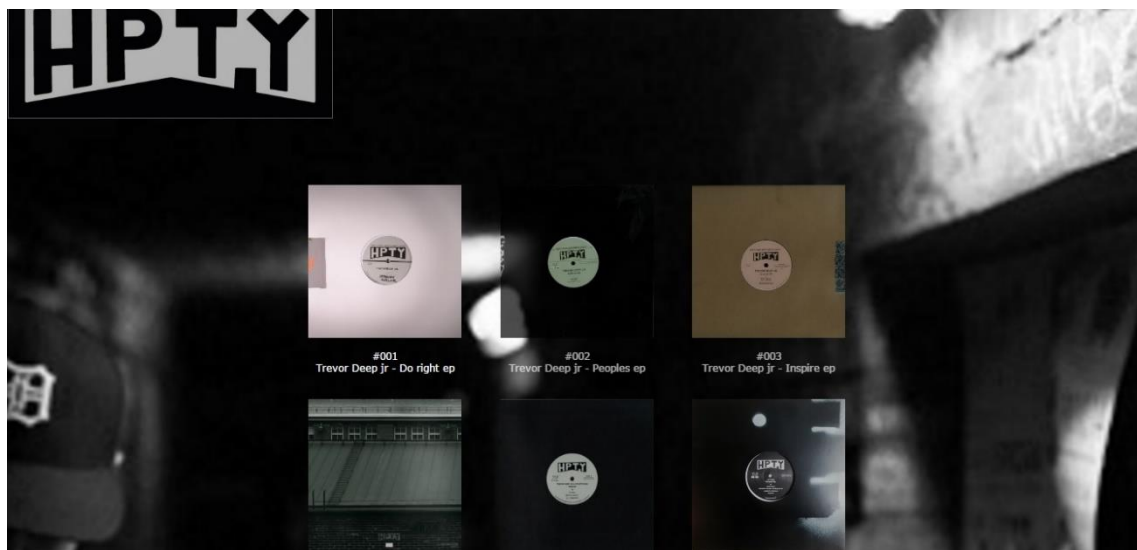
```

Figur 4 Kod för hem knappen.

För att publikationerna skulle spela huvudrollen på webbplatsen var det behändigt att använda också bilder av skivorna som knappar. Bilderna stod ut ganska starkt men genom att göra bilderna lite transparenta med kommandon **opacity: 0.7 filter: alpha(opacity=70);** och **.img:hover, .logo:hover {opacity: 1.0;filter: alpha(opacity=100);}** passade de in i bakgrundsbilden mycket bättre. Samma effekt finns på logon. Hover effekten gör att när man ställer musen på bilden försvinner genomsynligheten och bilden syns starkare, men effekten fungerar såklart inte på pekskärms apparater.

Placeringen av elementen på startsidan hade jag några förslag på: logon högst uppe i vänstra hörnet eller högst uppe i mitten och skivorna i en rad vertikal rad upp ifrån ner eller horisontell rad från vänster till höger. Kunden valde att logon skulle placeras högst uppe i vänstra hörnet och att skivorna skall vara i mitten av sidan.

På sidan skulle också finnas länkar till Facebook.com, Discogs.com och Soundcloud.com. Footern, som finns längst nere på sidan, var bästa alternativet för att placera länkarna i och finns på alla sidor på webbplatsen.



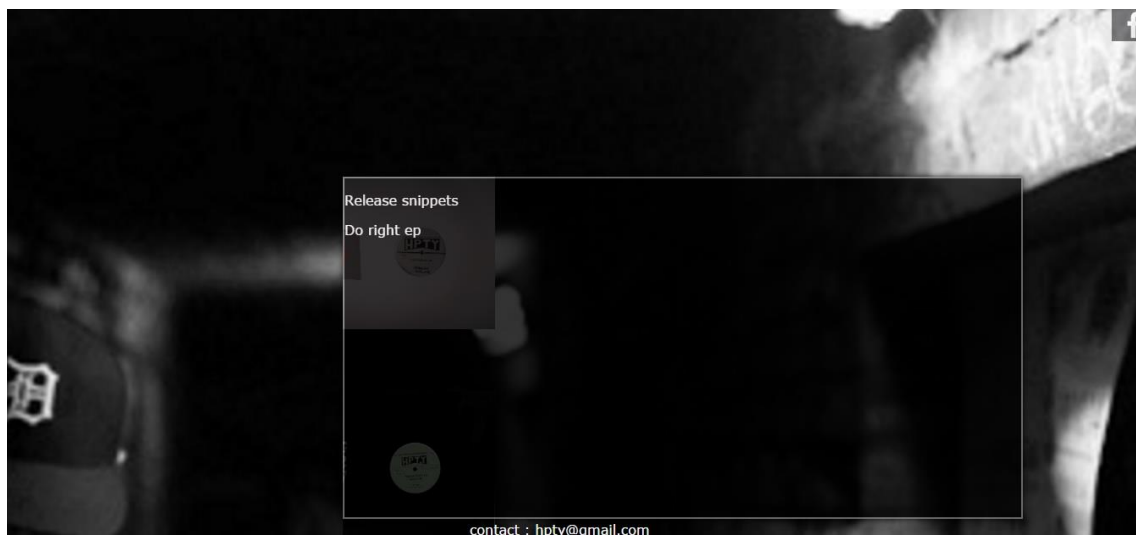
Figur 5 Slutliga designen på startsidan. Footern rymdes inte med på bilden.

Subsidorna skulle ha samma bakgrundsbild och logon skulle ha samma positionering och en iframe skulle spela huvudrollen på sidan. Iframen innehåller en musikspelare med snuttar av låtar från skivan ifråga. Låtsnuttarna finns på soundcloud.com och man kan bädda in musikspelaren enkelt via *soundcloud embed* koden som soundcloud.com bjuder på. Kunden ville också ha en youtube.com länk till skivans första låt. Länken valde jag att placera ovanför iframen. Iframen skall såklart spela huvudrollen på sidan och är placerad mitt på sidan.



Figur 6 Designen av subsidorna, endast iframens innehåll ändras.

Ett förslag som jag jobbade på, men kunden inte gillade, var ett modal fönster. Istället för att gå till en subsidia när man klickar på någon av skivorna skulle det öppnas en modal där iframen och musikspelaren skulle ha placerats.



Figur 7 Ett modal fönster som skulle innehålla information av skivan och en musikspelare. Ett förslag som jag hade.

```

10 <script type="text/javascript">
11
12
13 function showModal(index) {
14
15     var eps = [{
16         title: "Do right ep"
17     }, {
18         title: "Peoples ep"
19     }, {
20         title: "Inspire ep"
21     }, {
22         title: "TDJ lp"
23     } ];
24
25
26     var selectedEp = eps[index];
27     console.log(selectedEp);
28
29     $('#ep-title').text(selectedEp.title);
30
31     $('#modal').removeClass('modal-hidden');
32
33 }
34
35
36 function hideModal() {
37     console.log('hide modal');
38     $('#modal').addClass('modal-hidden');
39 }
40 </script>
41 <body>
42 <section>
43 <div id="header">
44 <a href="https://www.facebook.com/HPTVRecs">

```

Figur 8 Kod för modalen.

### 3 RESPONSIVE OCH ADAPTIVE WEB DESIGN

Responsive web design (RWD) och adaptive web design (AWD) betyder att webbsidan är byggd på ett sådant sätt att den anpassar sig enligt skärm storleken av apparaten, som används för att läsa webben, och ser bra ut på alla olika storlekars display. En webbsida som inte är responsiv eller adaptiv ser dålig ut på en mindre display och oftast mister webbsidan sin funktionalitet också. Genom att göra webbsidan responsiv/adaptiv kan problemen åtgärdas. Det finns flera olika sätt att göra detta.

Responsive web design och adaptive web design är två skilda saker, fastän man oftast talar om responsiv web design. De har ganska mycket likheter och med båda metoderna kan man nå ett bra resultat.

Vad som skiljer termen RWD från AWD är att en webbsida som har gjorts med RWD metoden anpassar sig flexibelt till skärmen oberoende vilken bredd eller storlek skärmen



har och använder också brytningspunkter till nytta. AWD anpassar sig också enligt skärmstorleken men gör det via i förväg bestämda brytningspunkter. Med AWD metoden kan man också göra ändringar i sidstrukturen med Javascript funktioner.

AWD förknippas ofta till termen progressive enhancement, som översatt till svenska betyder progressiv förbättring. Tanken är att webbsidan skall fungera på alla apparater och se bra ut, men för apparater med mindre display och sämre anslutningshastighet är webbsidan enklare och lättare. Ett vanligt exempel är att en bild ersätts av en text genom ”alt” attributen i HTML5 koden. D.v.s. ju större display desto snyggare är webbsidan och kan innehålla större bilder t.ex.. På mindre apparater visas alltså en avskalad version av egentliga webbsidan, men man strävar efter att funktionaliteten och innehållet är detsamma. (w3wiki, Graceful degradation versus progressive enhancement s.32)

### 3.1 RWD

Boken Responsive web design (Marcotte 2011) ger en bra översikt om vad responsiv design går ut på. Författaren av boken, Ethan Marcotte, nämner de tre viktigaste byggstenarna för en responsiv webbsida (Marcotte 2011 s.32):

1. A flexible, grid-based layout. (En flexibel layout som baserar sig på rutnät).
2. Flexible images and media. (Flexibla bilder och media).
3. Media queries, a module from CSS3 specification.

Med flexible grids och flexible images menar Marcotte att elementen på sidan är av storleken relativa och kan därmed ändra på sin storlek enligt fönsterstorleken. Resten sköts med media queries som också utnyttjas i AWD. Det är nästan omöjligt att göra en responsiv webbsida utan media queries.

När man strävar efter en snygg layout på sin webbsida, oberoende på vilken apparat man läser sidan med, är Marcottes modell en enkel och effektiv lösning. I hans modell anger man storleken av elementen i procent och em, som är relativa storlekar. T.ex. om en bild har storleken **width:20%**, betyder det 20% av bredden av elementet som bilden ligger i.

Genom att använda relativa storlekar istället för absoluta, som t.ex. pixlar, kan man åstadkomma en flexibel design som skalar sig bättre till olika storlekars display. Det är inte heller svårt att omvandla storlekar från absoluta storlekar till relativa. Ethan Marcotte använder sig av formeln ***target / context = result***. T.ex. 10px / 20px = 0,5 vilket är 50%.

Det är en bra början att för att nå en responsiv design. Men när man manuellt ändrar på storleken av webbläsarfönstret kan man se att elementen rör på sig och avstånden till varandra ändras och det uppstår fel. Punkter där designen går sönder kallas för brytningspunkter. Med hjälp av media queries kan man anpassa designen för en viss storleks skärm eller tillägga en helt ny design, om man så vill. Media queries brukar vara vid brytningspunkterna. En media query för en iPad med fönsterbredden 768px skulle se ut så här: ***@media screen and (min-width: 768px) and (max-width: 1024px) {...}***. Det betyder att när skärmens bredd är större än 768px och mindre än 1024px gäller CSS reglerna som finns inne i {...}. Media queryn placeras oftast inne i CSS filen men kan också placeras i HTML5 filens <head> tag. Exempel på en media query från mitt arbete finns nedan.

```
@media screen and (max-width: 768px) and (min-width: 501px){  
  .logo {  
    max-width: 400px;  
    width: 100%;  
    max-height: 250px;  
    width: 100%;  
  }  
  .logo img{  
    max-width: 400px;  
    width: 100%;  
    max-height: 250px;  
    height: 100%;  
  }  
  .img {  
    padding-left: 20%;  
  }  
}
```

Figur 9 En media query från HPTYs nätsida.

För att byta till en ny CSS-fil måste media querien göras i HTML koden innanför <head> taggen: ***<link rel="stylesheet" type="text/css" href="ipad.css" media="screen and***

*(min-width=768px) and (max-width=1024px)"/>*. Det går alltså att göra en ny eller anpassad CSS för alla fönsterstorlekar. En rad kod som berättar för webbläsaren vilken storleks display webbsidan skall anpassas för måste finnas med. *<meta name="viewport" content="width=device-width, initial-scale=1" />* är den populäraste metoden men det finns andra varianter också. Raden placeras i HTML5 kodens *<head>* del för att designen skall skalas för rätt storleks display.

För att inte göra det allt för enkelt måste man också ta i beaktande olika webbläsare och operativ system. Det som fungerar utmärkt i Google Chrome och Mozilla Firefox kanske inte fungerar lika bra i Safari och Internet Explorer. Internet Explorer 8 och äldre versioner stöder inte media queries. Det kan dock åtgärdas med hjälp av *respond.js* biblioteket eller *css3-mediaqueries.js* biblioteket.

### 3.1.1 Flexbox layout

Flexbox är en ganska ny modul som förenklar RWD en hel del. På CSS-tricks.com hittar man en guide för flexbox. På webbsidan beskriver man flexbox modulen såhär:

Huvudtanken bakom flex layouten är att ge containern förmågan att ändra på dess innehålls storlek (och ordning) och att på bästa möjliga sätt använda det tillgängliga utrymmet. Inom parent containern har man webbsidans innehåll i så kallade child items. När man ger parent containern värdet *display: flex;* får parenten makt över child items och kan därmed ändra på storleken och ordningen av child items, dock innehållet och marginalen i child items inverkar det inte på. När man använder flexbox slipper man också att använda kommandona *float*, *clear* och *vertical-align*. Flexbox försöker automatiskt fylla parent containern vertikalt eller horisontellt för att utnyttja all rum som finns i parent containern

.

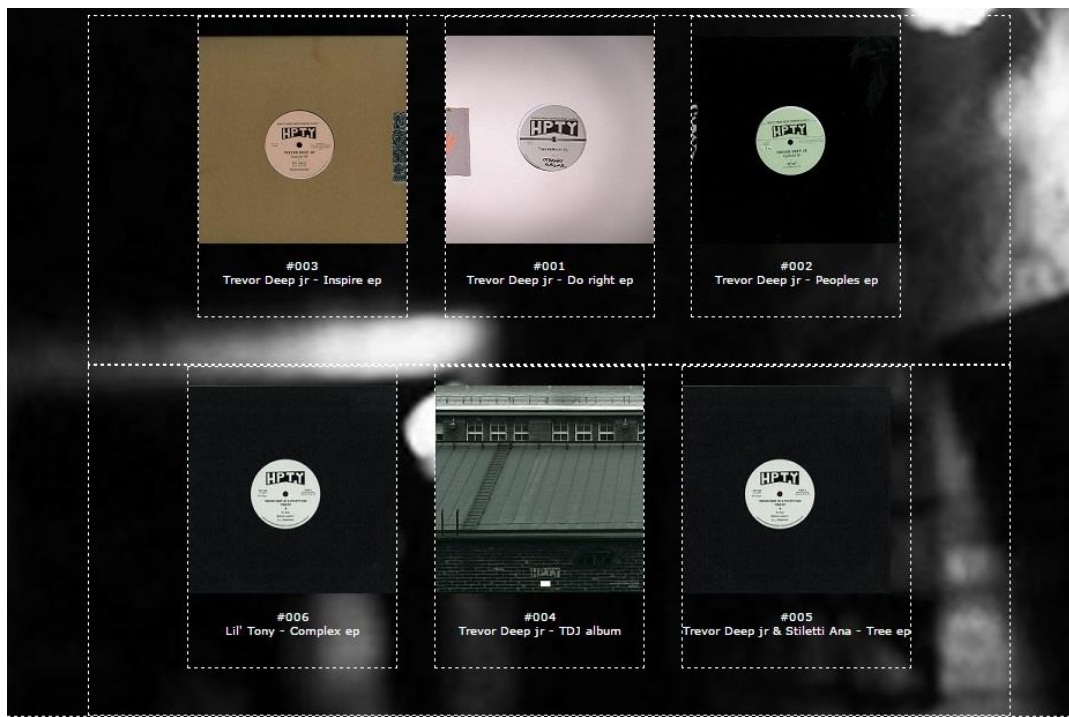
Med flexbox kan man också ändra på ordningen av child items till sitt eget tycke med kommandot *.item:nth-of-type(1) { order: 3; }*, om man vill. Flexbox har såklart andra mycket bra funktioner men de behöver jag inte för att nå mitt mål. Ett exempel från mitt arbete på hur flexbox beter sig:

```

17  div {
18      border: 1px dashed white;
19  }
20  .container {
21      display: flex;
22      flex-wrap: wrap;
23      position: center;
24      justify-content: center;
25      margin: 0 20% 0 20%;
26      padding-bottom: 3%;
27  }
28  }
29  .item:nth-of-type(1) { order: 3; }
30  .item:nth-of-type(2) { order: 4; }
31  .item:nth-of-type(3) { order: 1; }
32  .item:nth-of-type(4) { order: 5; }
33  .item:nth-of-type(5) { order: 2; }
34
35  .item {
36
37      display: flex;
38      max-width: 100%;
39      max-height: auto;
40      margin: 0 2% 0 2%;
41      padding: 2% 0 2% 0;
42      font-size: 80%;
43      overflow: hidden;
44      text-overflow: ellipsis;
45  }
46  .item img {
47      display: flex;
48      max-width: 180px;
49      width: 100%;
50      height: auto;
51  }

```

Figur 10 CSS för flexbox exempel



Figur 11 Parent container och sex child items.



*Figur 12 När displayn blir mindre flyttar parent container på child items.*

Flexbox är ett effektivt och enkelt sätt att ha kontroll över innehållet på websidan, men för mitt ändamål var flexbox inte den bästa lösningen. Font storleken påverkas inte. Det skulle behövas javascript eller media queries för att ändra på det. Samma gäller bilderna.

### **3.1.2 Bootstrap 3**

Bootstrap utvecklades av Twitter och den gjordes tillgänglig för allmänheten år 2011. Bootstrap är ett av de populäraste ramverken som används i utvecklingen av webbsidor. Bootstrap är också väldigt bra dokumenterat och Tack vare att Bootstrap utnyttjar sig av "fluid grids" är det väldigt enkelt att bygga en responsiv webbsida. Man kan placera upp till 12 kolumner horisontellt, varav alla är flexibla och anpassar sig enligt skärmstorleken.

Det fungerar lite på samma sätt som flexbox modulen, vilket kommer att stödas i Bootstrap 4. På webbsidan w3schools.com (Bootstrap get started) hittar man nödvändig information om Bootstrap. I fall man behöver javascript har Bootstrap färdigt inbyggda funktioner som är lätta att använda. För att kunna använda Bootstrap måste man endera ladda ner den som en fil från getbootstrap.com eller länka till CDN. För att Bootstrap skall fungera behövs några rader kod i <head> taggen, som kan se ut t.ex. så här:

```
<link rel="stylesheet"href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"> för att få Bootstrap CSS filen med,  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">  
</script> för att javascript delen skall fungera och ännu <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script> för att  
Bootstraps inbyggda javascript skall fungera.
```

Det finns dock några nackdelar med Bootstrap. På grund av att klasserna och komponenterna är färdigt byggda ser en stor del av webbsidorna, som har byggts med Bootstrap, likadana ut. Det märker man direkt när man surfar på webben. Det går såklart att finjustera Bootstraps färdiga klasser och ha en egen CSS. Bootstrap är också ganska tungt om man väljer att använda alla komponenter som är tillgängliga. Det kan leda till en ökad laddningstid.

### 3.1.3 Restive.js

Restive.js är en jQuery plugin som gör nästan all arbete för programmeraren, för att webbsidan skall vara responsiv/adaptiv. Restive.js har inbyggda jQuery kommandon som gör att man inte behöver använda media queries överhuvudtaget, utan allt sköts med javascript.

```
#my-element-id {/* Styles */} //my desktop style rule  
.phone #my-element-id {/* Styles */} //my general smartphone device style rule  
.tablet #my-element-id {/* Styles */} //my general tablet device style rule
```

Figur 13 Ett exempel hur en CSS3 fil kan se ut när man använder Restive.js. (Hämtad 20.4.2016. <https://speckyboy.com/2014/04/24/make-any-website-responsive/>)

```
#my-element-id {} //my desktop style rule

@media only screen and (min-width : 320px) and (max-width : 480px) {
/* Styles for some mobile devices */
}

@media only screen and (min-width : 321px) {
/* Styles for some other mobiles */
}

/** ...And yet more Media Query Directives **/
```

Figur 14 Hur det skulle se ut med media queries. (Hämtad 20.4.2016. <https://speckyboy.com/2014/04/24/make-any-website-responsive/>)

I stället för att göra ändringar i CSS klasserna via media queries skriver man nya CSS klasser för olika storlekar display. Elementens storlekar anges oftast i relativa storheter, som t.ex. procent eller em, men i vissa fall kan också använda absoluta storheter som pixlar användas.

```
/** Responsive CSS - Using Media Queries **/
@media only screen and (max-width: 320px) {
  /**phone in portrait CSS rules here**/
  section {width: 100%; padding: 0 10px;}
  header #logo div {font-size: 320%; text-align: center; margin: 0 auto;}
  header #menu-container {height: auto;}
  header #menu nav ul {height: auto;}
  header #menu nav ul li {float: none; width: auto; height: 40px;
    line-height: 40px; border-bottom: 1px dashed #444;}
  #content #main{float: none; width: 100%;}
  #content #sidebar{display: none;}
}

@media only screen and (min-width : 321px) and (max-width : 480px) {
  /**phone in landscape CSS rules here**/
  section {width: 100%; padding: 0 10px;}
  header #logo div {font-size: 320%; text-align: center; margin: 0 auto;}
  header #menu-container {height: auto;}
  header #menu nav ul {height: auto;}
  header #menu nav ul li {float: none; width: auto; height: 40px;
    line-height: 40px; border-bottom: 1px dashed #444;}
  #content #main{float: none; width: 100%;}
  #content #sidebar{display: none;}
}
```

Figur 15 Hälften av CSS:n som behövs för skriva media queries för alla storleks display. (Hämtad 20.4.2016. <https://speckyboy.com/2014/04/24/make-any-website-responsive/>)



```

/** Responsive CSS - Using Restive.js */
.mobi section {width: 100%; padding: 0 10px;}
.mobi.phone header #logo div {font-size: 320%; text-align: center; margin: 0 auto;}
.mobi.tablet header #logo div {font-size: 360%; text-align: right;}
.mobi.phone header #menu-container, .mobi.tablet.portrait header #menu-container {height: auto;}
.mobi.phone header #menu nav ul, .mobi.tablet.portrait header #menu nav ul {height: auto;}
.mobi header #menu nav ul li {float: none; width: auto; height: 40px; line-height: 40px; border-bottom: 1px dashed #444;}
.mobi.tablet.landscape header #menu nav ul li {float: left; width: 20%; height: 60px; line-height: 60px;}
.mobi #content #main{float: none; width: 100%;}
.mobi.tablet.landscape #content #main{float: left; width: 65%;}
.mobi #content #sidebar{display: none;}
.mobi.tablet.landscape #content #sidebar{display: block; float: left; width: 33%; padding: 0 0 2%;}

```

Figur 16 CSS som krävs för att webbsidan skall vara responsive på alla olika storleks display, t.o.m. både i portrait och landscape. ((Hämtad 20.4.2016. <https://speckyboy.com/2014/04/24/make-any-website-responsive/>)

Det behövs alltså mycket mindre CSS kod när man använder restive.js. Huvud poängen är att spara tid. Restive.js utnyttjar också brytningspunkter. Brytningspunkterna matas in i funktionen som sedan väljer rätta CSS klasser för rätt storleks display.

```

<script>
$(document).ready(function () {
    $('body').restive({
        breakpoints: ['240', '320', '480', '640', '960', '1024', '1280'],
        classes: ['css-240', 'css-320', 'css-480', 'css-640', 'css-960', 'css-1024', 'css-1280']
    });
});
</script>

```

Figur 17 En function där det finns brytningspunkterna och namnen på klasserna som man vill att skall användas i vid dessa brytningspunkter. (Hämtad 20.4.2016 <http://docs.restivejs.com/>)

Restive.js verkar alltså vara en bra lösning för att göra en webbsida responsiv. På grund av att det är en plugin kommer det att öka på fil storleken, men jämfört med t.ex. Bootstrap är restive.js ganska litet. Webbsidans laddningstid ökar antagligen lite men inte så mycket, vid användning av restive.js.



## 4 GENOMFÖRANDET

Kodandet av websidan orsakade inga problem. Webbplatsen består av totalt sju sidor HTML5 kod och två sidor CSS kod. Att finslipa layouten krävde mest tid och var i mellan åt nervprovande, men inte problematiskt.

### 4.1 Verktyg

För själva kodandet behövde jag ett textredigeringsprogram. Det finns många att välja mellan och jag valde Sublime text 2. Sublime har många bra egenskaper. För bildredigering valde jag att använda Photoshop CS5.

För att kunna testa webbsidans responsivitet har jag använt Samsung galaxy S4, Sony Vaio 13", Asus Google Nexus 7". För att kunna testa sidan på Safari och iOS måste jag vända mig till min kontaktperson som sedan testade webbsidan med iPad och iPhone5 och skickade screenhots till mig. Jag letade också på internet efter olika iOS emulatorer och hittade flera, men jag tvivlar på att de skulle fungera som de ska. Med Apples Xcode skulle det säkert ha fungerat men jag hade inte tillgång till en Mac dator.

Efter att designen var klar och hade fått godkännande av kunden hade jag redan jobbat en hel del på själva kodandet av sidan. Jag designade alltså sidan rakt med HTML5 och CSS istället för att först göra en skiss med Photoshop och sen sätta igång med själv kodandet. Jag tror att det sparade jag lite tid på, fastän jag måste göra mycket förändringar i CSS filerna efteråt för att jag hade använt absoluta värden på allt.

#### 4.1.1 Hastighets test

Webbplatsen har inte så mycket innehåll men bilderna och iframen gör den ganska tung och laddningstiden är ganska lång. Jag satt upp webbsidan på Arcadas server för att kunna testa laddningstiderna. Testerna gjorde jag på webbadressen [tools.pingdom.com](http://tools.pingdom.com) som är gratis till en viss punkt. På sidan får man ganska mycket information om testresultatet och av webbsidan som man har testat.



Figur 18 Resultat av hastighetstest för startsidan.

Fastän sidan består nästan bara av bilder laddar sidan ganska snabbt.



Figur 19 Hastighetstest för en subsida.

Av testerna kan man läsa att subsidan laddar betydligt långsammare fastän innehållet på sidan är nästan hälften mindre. Av detta kan man dra slutsatsen att det är iframen, som innehåller musikspelaren, ökar på laddningstiden märkvärdigt. Detta beror på att den hämtas från Soundcloud.com.

## 4.2 Diskussion och motiveringar

När jag började arbetet hade jag bara lite kunskap om metoder som utnyttjas för att nå responsivitet. För att kunna hitta en bra lösning för just denna webbplats måste jag studera lite djupare i ämnet. För att kunna fördjupa sig i ämnet måste man veta grunderna. Därför läste jag Ethan Marcottes bok: Responsive web design. Den gav en bra översikt vad responsivitet går ut på.

Bootstrap visste jag om redan tidigare och det var därför mitt första alternativ. Media queries var inte heller något alldeles nytt för mig, men flexbox och restive.js hade jag inte hört av tidigare, men de verkade vara bra lösningar. Eftersom jag hade gjort designen rakt med HTML5 och CSS på min laptop kunde jag glömma mobile first metoden. Det kunde ha varit en bra metod just för detta arbete. Att börja från den minsta skärmstorleken och sedan anpassa webbsidan till större display kunde ha varit en enklare tillvägagångssätt.

Jag valde dessa metoder för närmare undersökning. Sedan satte jag igång och tillämpade metoderna.

När jag påbörjade arbetet funderade jag på att använda Bootstrap för dess goda egenskapers skull och för att vem som helst kan vidareutveckla webbsidan i framtiden. Varför jag ändå valde att inte använda Bootstrap var uppbyggnaden och designen av sidan. Designen skulle vara simpel och består av bara några komponenter per sida. Jag behövde inga navigeringsfält, inga kolumner med text och ingen javascript. Egentligen skulle jag inte ha haft någon nytta av Bootstraps färdigt byggda komponenter. Bootstrap är ett bra verktyg men skulle inte vara så nyttig för denna webbplats.

Flexbox var ett jätte bra alternativ för mitt ändamål. Att få en horisontell rad med bilder att automatiskt anpassa sig till en vertikal linje, när webbläsarfönstret blir mindre, har aldrig varit lika enkelt. Det krävdes dock ändå media queries för att innehållet skulle ha sett bra ut på en mindre skärmar.

Restive.js tillämpade jag inte själv till mitt arbete. På grund av att webbsidans laddningstid redan är så hög ville jag undvika javascript. Jag läste dock noggrant igenom ett exempel på restive.js dokumentationssida och kom till slutsatsen att det är en bra lösning men inte för en liten webbplats, som denna.

Jag funderade länge på att istället använda flexbox modulen men beslöt att använda Ethan Marcottes modell. Orsaken varför jag valde att använda Ethan Marcottes modell är att den fungerar garanterat, den är lätt att förstå om någon annan vill vidareutveckla sidan och den kräver inte så mycket kod.

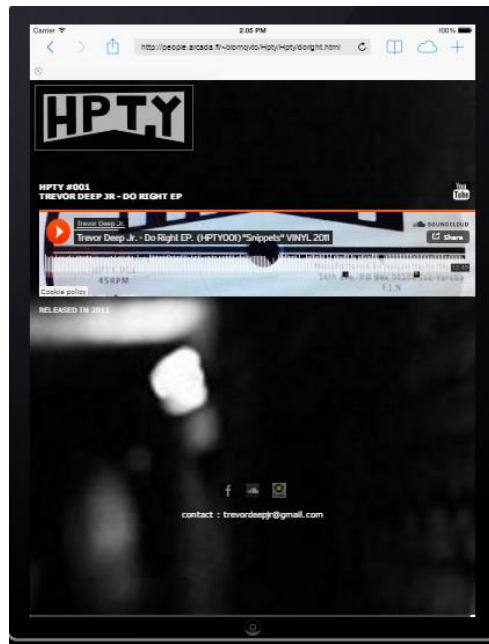
Jag tror inte att det finns en överlägset bästa lösning till att göra just denna webbplats responsiv. Det skulle säkert ha lyckats bra med alla dessa metoder men Marcottes modell och flexbox modulen skulle jag rekommendera.

Jag tycker att jag lyckades nå ett ganska bra resultat med de färdigheter som jag hade inom ämnet och de verktyg som jag hade till mitt förfogande. I processen lärde jag mig hela tiden något nytt och det krävdes också för att kunna nå ett bra resultat.

Vad som tillfredsställer mig mest är att kunden är väldigt nöjd med webbplatsen. De har redan frågat om jag skulle vilja fortsätta och vidareutveckla webbplatsen i framtiden.



Figur 20 Webbssidan på laptop.



Figur 21 Webbstan på en iphone5 och ipad.

## KÄLLOR

Marcotte, Ethan. 2014, *Responsiv web design*, 2uppl., New York: A Book Apart, 153 s.

### Elektroniska källor

10 Best reasons to use bootstrap for amazing web designs. Tillgänglig <https://www.devsaran.com/blog/10-best-reasons-use-bootstrap-amazing-web-designs>

hämtad 15.4.2016

Flexbox documentation. Tillgänglig <https://css-tricks.com/snippets/css/a-guide-to-flex-box/> hämtad 23.4.2016

Bootstrap get started. Tillgänglig [http://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](http://www.w3schools.com/bootstrap/bootstrap_get_started.asp) hämtad 23.4.2016

Testandet av designen på ipad. Tillgänglig <http://ipadpeek.com/>

hämtad 20.4.2016

Restive.js dokumentation. Tillgänglig <http://docs.restivejs.com/> Hämtad 20.4.2016

Responsive web design. Tillgänglig <http://learn.shayhowe.com/advanced-html-css/responsive-web-design/>

Hämtad 20.4.2016

Restive.js dokumentation. Tillgänglig <http://restivejs.com/>

Hämtad 20.4.2016

Apparaters storlekar. Tillgänglig <http://mydevice.io/devices/>

Hämtad 20.4.2016

How to make any website responsive. Tillgänglig <https://speckyboy.com/2014/04/24/make-any-website-responsive/>

Hämtad 20.4.2016

Graceful degradation versus progressive enhancement. Tillgänglig  
[https://www.w3.org/wiki/Graceful\\_degradation\\_versus\\_progressive\\_enhancement](https://www.w3.org/wiki/Graceful_degradation_versus_progressive_enhancement)

Hämtad 20.4.2016

Hastighetstest. Tillgänglig: [www.tools.pingdom.com/fpt/](http://www.tools.pingdom.com/fpt/)

Hämtad 25.4.2016

